Software Requirements Specification

for

Automated Lecture Hall Booking Portal

Version 0.1

Prepared by-

Group 12 **Group Name: Aviators**

#	Name	Roll No.	Email
•	Chaitanya Bramhapurikar	230305	bcvishwas23@iitk.ac.in
•	Rahul Meena	230832	rahulkcm23@iitk.ac.in
•	Bhavya Chauhan	230294	bhavyach23@iitk.ac.in
•	Atharv Phirke	230238	atharvp23@iitk.ac.in
•	Aaradhya Rohi	230011	aaradhya23@iitk.ac.in
•	Chaudhari Divyesh	230325	divyesh23@iitk.ac.in
•	Devansh A Dhok	230354	devanshad23@iitk.ac.in
•	Areen Mahich	230188	areenm23@iitk.ac.in
•	Daksh Dua	220321	dakshdua22@iitk.ac.in
•	Avantika Rohite	230245	avantikar23@iitk.ac.in

Course: CS253
Mentor TA: Souvik Mukherjee
Date: 24 January 2025

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Group 12	Initial Draft	24/01/25

1 Introduction

1.1 Product Scope

Our product, an **"Automated Lecture Hall Booking Portal"** is designed to streamline and simplify the process of booking lecture halls for various purposes in IIT Kanpur. It allows faculties, administrators and other authorized personnel to efficiently manage Lecture Hall bookings. This system drastically reduces manual efforts in lecture hall bookings, also addressing issues such as manual error, double bookings and scheduling conflicts. Users can view real-time availability of a lecture hall for all the time slots, request bookings and receive quick confirmation. The system's scope extends to features such as role-based access, user authentication and advanced search options for filtering lecture halls based on capacity, equipment and other features.

1.2 Intended Audience and Document Overview

This document is intended for **developers**, **professors** and **TAs** evaluating the same for the course project, and **end-users**, with the last category of audience likely to be one of the below-mentioned categories:

- **Professors:** Professors can use this portal for booking lecture halls for their course lectures and labs(for the entire semester), Quizzes, Make up classes and talks from researchers and other professors of the same or other universities to showcase their research work.
- **Clubs/Societies/Fests:** Heads of clubs, societies, fests and cells may also use this portal to book lecture halls for their events for the campus community.
- **Lecture Hall Administration:** They shall be the ones coordinating between the institute bodies(DoAA/DoSA) and the above two mentioned here, and cancel/approve booking requests based on availability.

Developers, professors and TAs should have a look at the sections 2, 3 and 4 of the document in order to have a strong understanding of the technical aspects of the portal, along with the functionality. On the other hand, end-users as elaborated above, just need to go through sections 2.1 and 2.2, just to understand the functioning of the overall software. Everyone reading this document must refer to 1.3 and the Appendix in case they encounter terms unfamiliar to them.

1.3 Definitions, Acronyms, and Abbreviations

Acronym	Description
ACID	Atomicity, Consistency, Isolation, and Durability
API	Application Programming Interface
AWS	Amazon Web Services
CSV	Comma Separated Values
DB	DataBase
DBMS	DataBase Management System
HTTPS	HyperText Transfer Protocol Secure
IT	Information Technology
JS	JavaScript
LHC	Lecture Hall Complex
MongoDB	Humongous DataBase
PDF	Portable Document Format
REST	Representational State Transfer
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language

Term	Definition
Admin	Administrative. The term ‘admin’ refers to the IIT-K Lecture Hall Complex Office staff. This type of user has full control over the system’s operation.
User	The term ‘User’ refers to the various entities/organizations in the IIT-K campus that need to request a Lecture Hall booking. Examples include Professors, Clubs, Student Placement Office.
Web-app	A software application that runs on a web browser and can be accessed from any device with an internet connection.

1.4 Document Conventions

The document adheres to the following font specifications to maintain consistency and readability:

1. **Base Font:**
    - The default font for the document is **Arial** with a base font size of **12pt**.
2. **Title Font Sizes:**
    - **Document Title:** Set in `\Huge`, approximately **24.88pt**.
    - **Subtitles (e.g., ”for,” ”Version 0.0”):** Set in `\Large`, approximately **14.4pt**.
3. **Section Titles:**
    - **Section Titles:** Set in **16.59pt**.
    - **Subsection Titles:** Set in **14.4pt**.
    - **Sub-subsection Titles:** Set in **12pt**.
4. **Other Text Elements:**
    - **Prepared By:** The ”Prepared By” label is set in `\Large`, approximately **14.4pt**.
    - **Tables and Lists:** All text within tables and lists is set to the base font size of **12pt**.
5. **Table of Contents:**
    - The Table of Contents title and entries use the base font size of **12pt**.
6. **Revisions Table:**
    - All text within the Revisions table is set to the base font size of **12pt**.

These font settings are designed to ensure a professional and cohesive appearance throughout the document.

1.5 References and Acknowledgments

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

We’d also like to acknowledge the help of our TA, **Mr. Souvik Mukherjee**, for their valuable input in the creation of this document. We also would like to thank **Prof. Indranil Saha** for teaching us the concepts of the course and guiding us.

2 Overall Description

2.1 Product Overview

The Automated Lecture Hall Booking Portal is a new, self-contained system designed to manage the scheduling and booking of lecture halls at IIT Kanpur. It is not a replacement for any existing system but an enhancement to current manual processes. The portal digitises the scheduling process and centralizes the user authentication system for secure role-based access. Users such as faculty, administrators, and other authorized personnel interact with the system to view lecture hall availability, book halls, and resolve scheduling conflicts. The system ensures efficient resource utilization by minimizing manual errors and efforts.

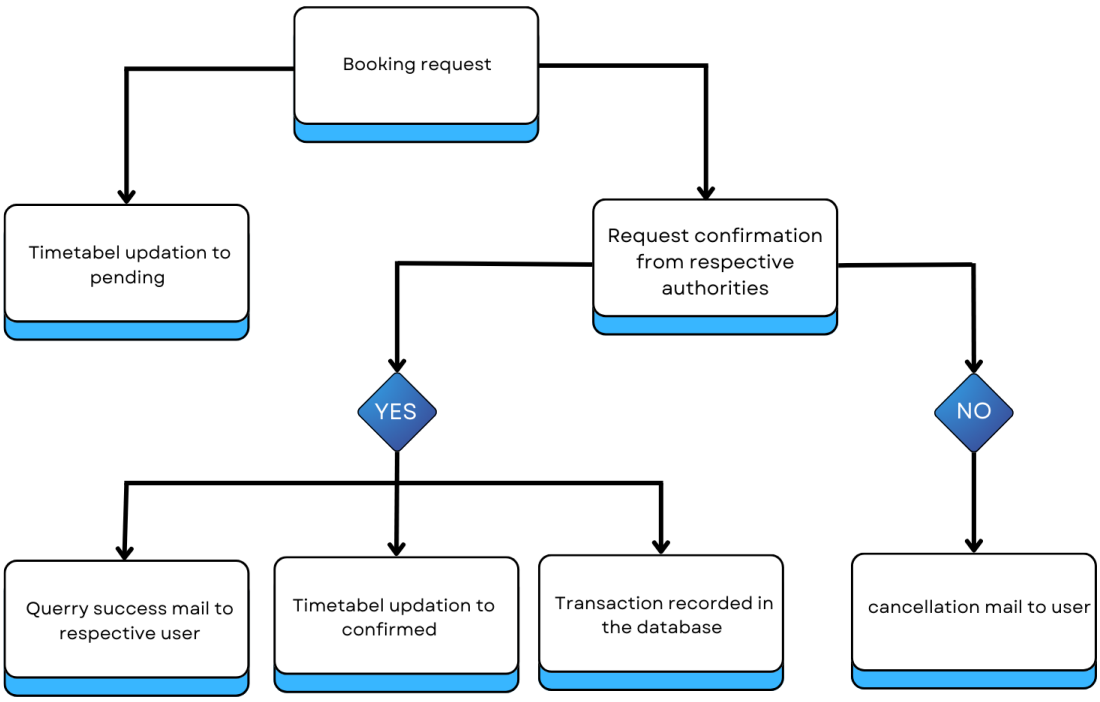

Figure 1: Flow Diagram

2.2 Product Functionality

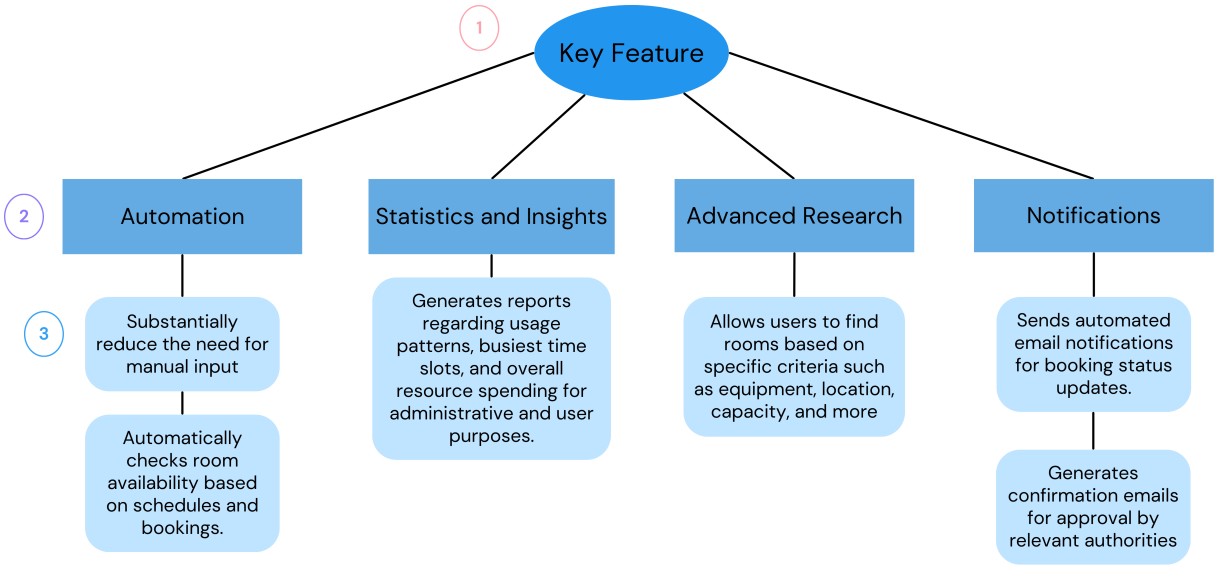

Figure 2: Product Features

2.3 Design and Implementation Constraints

- **Email Notification Service:**
  - The system depends on IITK webmail or another SMTP service for sending notifications. Any service downtime, misconfiguration, or traffic surge could lead to communication issues. Retry mechanisms and error logging should be implemented to handle failures effectively.
  - Scalability is required to handle high-traffic periods, such as during Semester Examination periods
- **Integration with Institute's academic Calendar:**
  - The portal might need to sync with the institute's academic calendar to avoid scheduling conflicts with pre-planned events and to prevent bookings on institute holidays.
- **Performance and Resource Management:**
  - Large report generation requires careful resource optimization to avoid overloading server CPU, RAM, and storage.
  - The system should ensure minimal delays while processing or querying data for reports, even under high server loads.
- **Browser and Device Compatibility:**
  - The application must be compatible with modern web browsers (e.g., Chrome, Firefox, Edge). Features like JavaScript and cookies are essential for functionality, and support for outdated browsers will not be provided.

- The design should also ensure **responsiveness** for different screen sizes, including desktops, tablets, and smartphones, if required.

- **Data Security and Integrity:**
  - User data and reports must be protected against unauthorized access. Encryption should be used for sensitive data in transit and at rest.
  - The system must adhere to IIT Kanpur's IT security and data handling policies. Sensitive information like passwords and financial records must be stored securely using encryption.

- **No Sign-Up Feature:**
  - There will be no user self-registration; credentials must be created and distributed manually by the LHC Office Staff(admin).

- **Interface Design:**
  - The system must provide two separate interfaces: one for end-users (professors and club coordinators) and another for administrators (LHC office staff). The interface should be accessible and responsive on both desktop and mobile devices.

- **Hardware Dependency:**
  - The system should integrate seamlessly with the LHC office's existing hardware and software systems.

2.4 Assumptions and Dependencies

2.4.1 Assumptions

- **Database Availability:** The relational database (e.g., MySQL, PostgreSQL) will remain operational with minimal downtime and will scale to handle data growth as the system expands.
- **Email Services:** IITK webmail or another configured SMTP service will handle notification requests efficiently, even during periods of high traffic. Configuration changes will be minimal and well-documented.
- **Valid Input Data:** Users will input data in the correct format, reducing validation errors during report generation and ensuring smooth operation.
- **Network Stability:** The system assumes stable and reliable network connectivity to facilitate interactions between the database, SMTP service, and user interface components.

2.4.2 Dependencies

- **Database Technology:** The system relies on a well-configured and operational database (MySQL, PostgreSQL) to store and retrieve data efficiently. Any downtime or misconfiguration could disrupt report generation and data consistency.
- **SMTP and Webmail Services:** The SMTP service, including IITK webmail, is critical for sending notifications. Failures, misconfigurations, or changes in these services could disrupt communication.
- **Authentication System:** Integration with institutional authentication systems is a core dependency. Changes in authentication protocols or system failures can block access for legitimate users.
- **Error Logging and Monitoring:** Dependencies exist on logging mechanisms to track and resolve database connection issues, SMTP failures, and report generation errors promptly. These logs should be easily accessible to administrators.
- **Server Resources:** Sufficient server resources (CPU, RAM, storage) must be available to support simultaneous requests for report generation, data queries, and notifications without performance degradation.
- **Network Connectivity:** Stable and fast network connectivity is required for real-time interactions between system components and to maintain uptime.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The software application will provide intuitive and responsive interfaces accessible via web browsers and mobile devices. The User and Admin will have different roles while also sharing some elements of the functionalities and interface. Below are the graphical views of the User and Admin interfaces.

I. **Login Screen:**

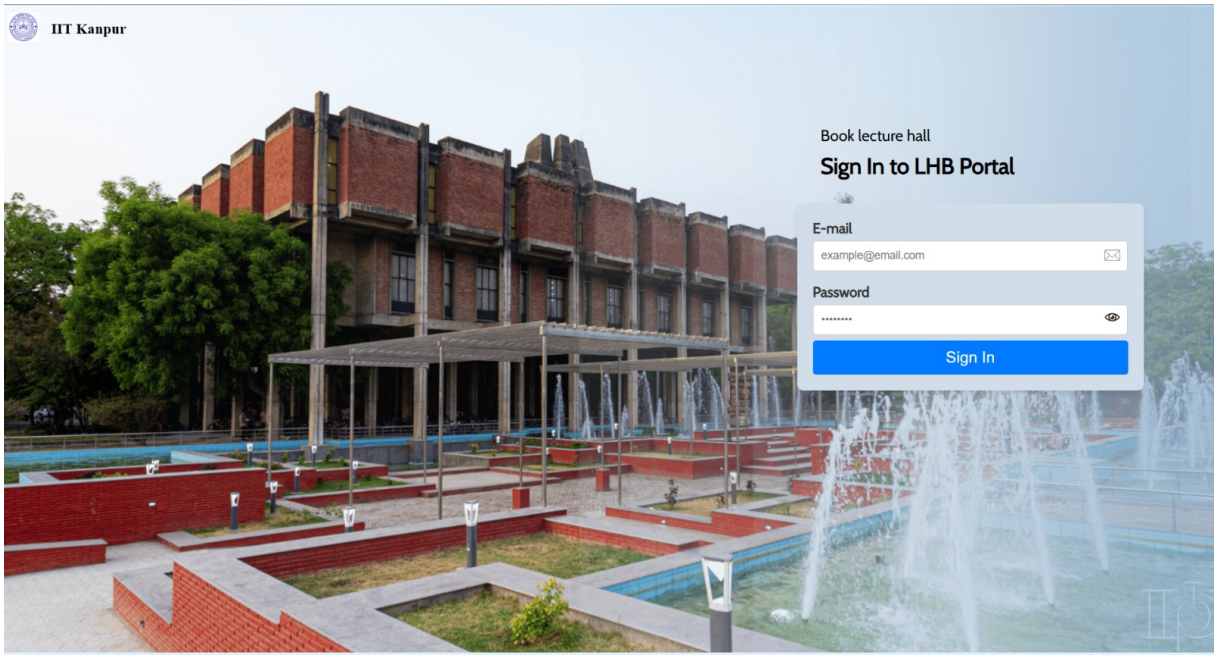

Figure 3: Login Screen

- **Description:**
  - The login screen will allow users and admin to authenticate securely.
  - Users and admin will enter their username and password.
  - Buttons for “Sign In” will be provided.
  - Validation messages will be displayed for incorrect credentials.
- **Interaction:**
  - Text fields for input.
  - “Sign In” button to process login.

II. **Dashboard:**

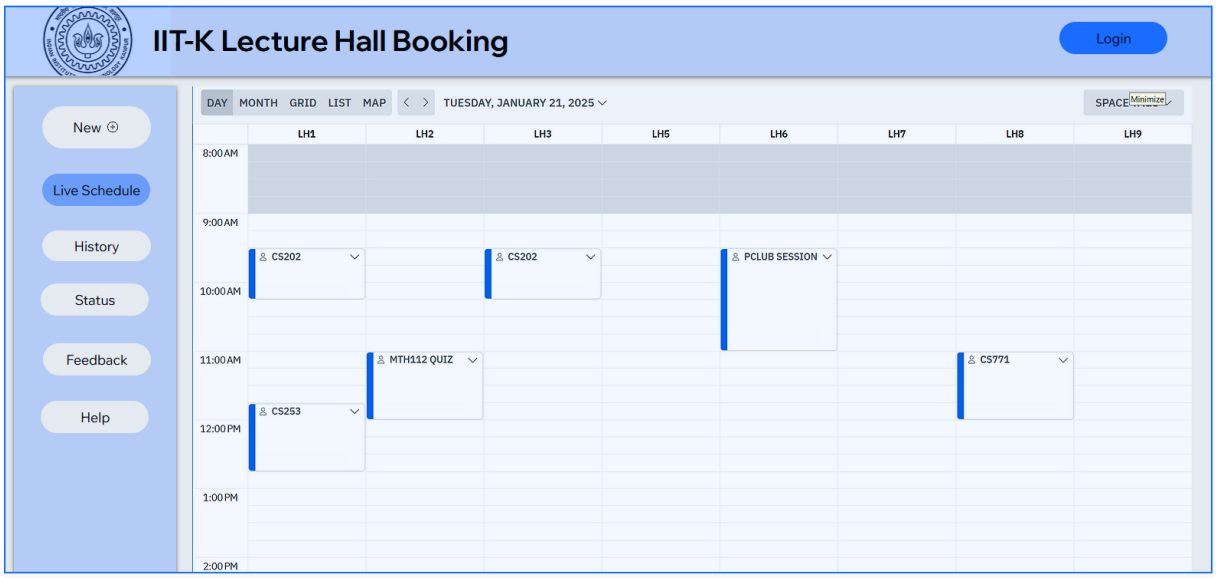

Figure 4: Dashboard

- **Description:** The dashboard serves as the central hub for user interactions based on their roles. The Live Schedule is the common element between both the User and Admin interface.
  - User Role: Options to submit a new request, view the status of previous requests, and access notifications, provide feedback.
  - Admin Role: Options to approve/reject requests, delete and view/search verification history.
- Interaction:
  - Live Schedule
  - Drop-down menus, clickable cards, and quick navigation buttons.
  - Notifications area for updates.

III. **History Tab - User:**

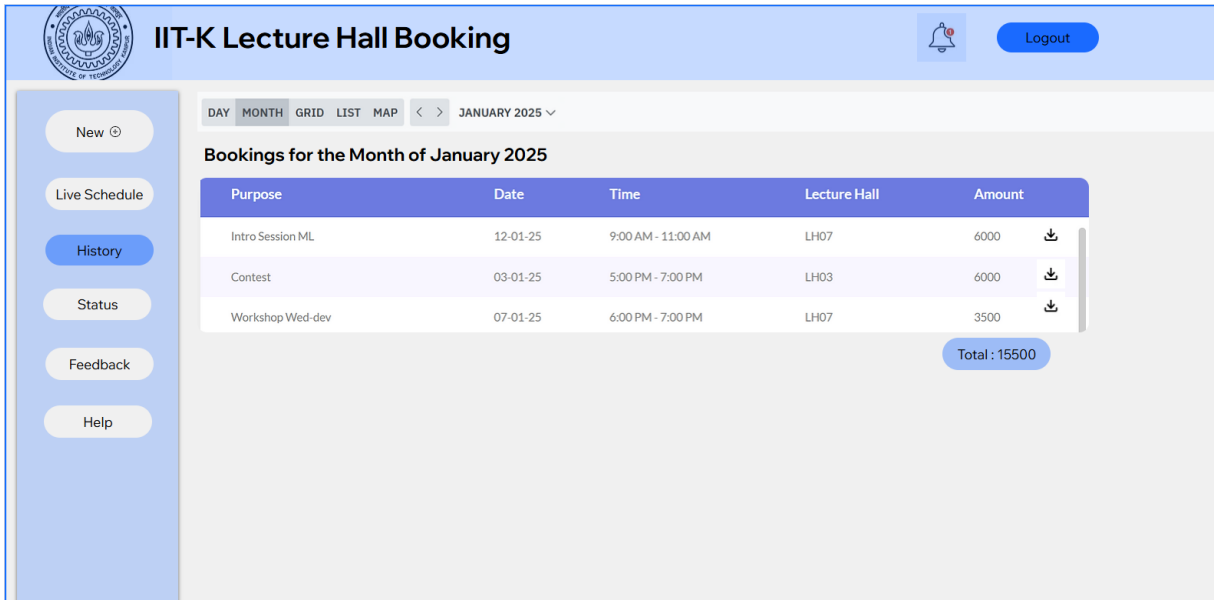

Figure 5: User History

- **Description:**
  - The History Tab allows users to view their past bookings.
  - The Auto-Generated Bills can be downloaded from here.
  - A table view displaying booking details such as: Purpose, Date, Time, Lecture Hall number and Amount.
- **Interaction:**
  - A search bar to filter bookings by date or month.
  - Dropdown filters for easy navigation.
  - An option to download bills for individual bookings.

IV. **Request Submission Form:**

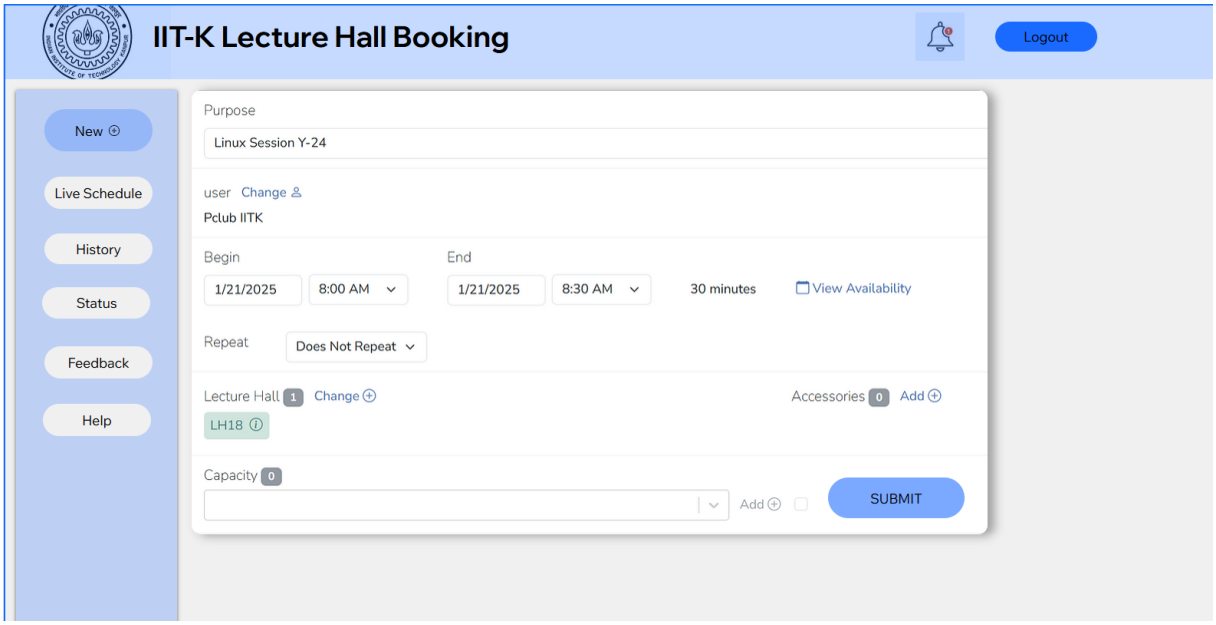

Figure 6: Request Submission Form

- **Description:**
  - After filling the form, the user can either:
    - Select a particular Lecture Hall, or
    - Let the system select it automatically.
  - Real-time validation to ensure form completeness.
- **Interaction:**
  - Text fields for: “Purpose”, “Capacity”
  - Input fields for: “Date”, “Lecture Hall”, “Accessories”
  - ”Submit” button for finalizing the request.
  - Interactive form fields with tooltips for assistance.

V. **Admin Interface - History Tab**

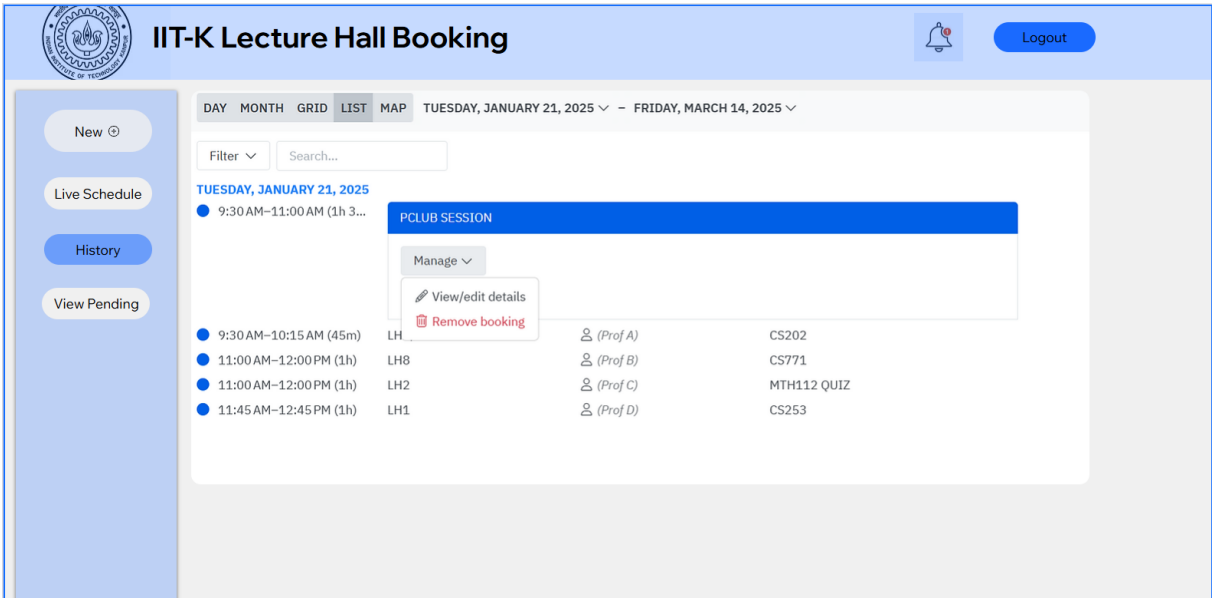

Figure 7: Admin Interface

- **Description:**
  - This interface is only accessible by Admin.
  - Authorities and Office Staff will access requests via a list view.
  - Each request entry will display essential details, such as: Status, Timestamps and Remarks
  - Actions like: “View/edit details”, “Remove bookings” will be available for each entry.
- **Interaction:**
  - Buttons to execute approval/rejection actions.
  - A Search bar to search for bookings.
  - Filters for date, month, Location

3.1.2 Hardware Interfaces

The application will **primarily** be a **web application**, with the possibility of extending to a mobile application if time permits. Therefore, the hardware interfaces involved include:

- **User Devices**
  - **Personal Computers (PCs):** Users will interact with the application through their PCs to access features such as: submitting requests, checking statuses and managing their accounts
  - **Mobile Phones:** If a mobile application is developed, users will access similar features on their smartphones, ensuring convenience and mobility.
- **Lecture Hall/Office Staff Devices**

- **Staff PCs:** Office staff will use their PCs to: Verify user requests Manage approvals Perform administrative tasks related to the system

- **Authority Devices**
  - **PCs:** Authorities will use their computers to review and process user requests submitted via the system
  - **Mobile Phones:** Through the mobile application, authorities can also accept or reject requests on the go, ensuring flexibility and efficiency.

3.1.3 Software Interfaces

- **Database Management System (DBMS)** The application will utilize a database to store and manage data efficiently. This includes:
  - **User Data:** Profiles, credentials, and role-specific information.
  - **Request Records:** Details of user-submitted requests, their statuses, and timestamps.
  - **Verification Logs:** Logs of actions taken by the office staff and authorities for tracking purposes.

  A relational database (e.g., MySQL, PostgreSQL) will be used based on project requirements.

- **Mail Client for Automatic Email Generation** The application will connect to a mail client or SMTP service (e.g., Gmail SMTP, Microsoft Outlook, or SendGrid) to automate email communication.
  - **Verification Emails:** Automatically sent to authorities for request verification.
  - **Notifications:** Updates to users about request status changes or approvals.

- **Authentication Services**
  - Integration with institutional authentication systems for specific user roles, such as office staff and authorities.

- **Web Server and API Services**
  - A web server will host the application and serve client requests.
  - RESTful APIs will facilitate communication between the frontend, backend, and external systems like databases or mail clients.

- **Frontend Framework**
  - The application will use Next.js, a React-based framework, for building the user interface.

3.2 Functional Requirements

3.2.1 User Authentication

- The system shall allow users (club coordinators, professors, and LHC employees) to log in using unique credentials, which will be provided by the admin.
- No “Sign up” or “Register” option will be provided, and all credentials shall only be created by LHC Office Staff.
- The system must ensure secure authentication using HTTPS and hashed passwords.

3.2.2 LH Booking

- The system shall provide users with an interface to select a date, time, LH, and capacity for booking.
- The system shall automatically identify user role based on their unique id and provide them with special permission and options. For example. A club coordinator shall not be able to book a LH for a Course Quiz or Exam.
- The system shall allow users to select additional equipment such as projectors, microphones, and blackboards.
- The system should automatically validate the availability of the selected hall and equipment before confirming the booking.
- Once requested, the system shall automatically send authentication mail to all required authorities for approval. Each authority shall receive a unique link to approve or reject the booking.
- The action of approving and rejecting requests shall be done by clicking on the link.
- All pending requests shall be auto-rejected after a specified time frame of 48 hours.
- The user should be able to select and request multiple rooms at once.
- The system shall notify the user of the approval/rejection status via mail once all authorities respond.

3.2.3 Timetable

- The system shall display a timetable view on the home page (as previously shown in 4) showing the schedule of LH for a given date.
- The timetable shall include details such as the time slot, the name of the booking user, and the purpose of the booking.
- This shall be visible to all visitors of the web-app.

3.2.4 Automated Invoice Generation

- Once the booking is confirmed by all the authorities, the system shall automatically generate an invoice for the booking, which will include details of hall usage, equipment charges, and any additional costs.

3.2.5 Notifications

- The system shall send email notifications to users upon successful booking, approval, or rejection.
- The system shall notify the admin in case of conflicts or errors during booking.

3.2.6 Administrative Functions

The system shall provide an admin interface for the LHC office staff to:

- View all bookings and their statuses.
- Add, update, or delete bookings.
- Generate monthly and annual reports for usage statistics of equipment and LHs.

3.3 Use Case Model

3.3.1 Use Case #1: User Submits Request (U1)

Author – Chaitanya, Rahul

Purpose - To allow users to submit a request through the web or mobile application. The request will be stored in the database and sent to authorities for verification.

Requirements Traceability –

- R1: The user must be able to log in securely.
- R2: The system must allow the user to submit a request form.
- R3: The request must be stored in the database and trigger email notification to authorities.

Priority - High – This is a core function of the application.

Preconditions -

- The user must be logged into the system.
- The request form must be valid and complete.

Post conditions -

- The request is stored in the database.
- An email notification is sent to the appropriate authorities.
- The user is notified of successful submission.

Actors –

- Primary Actor: User (Club-coordinator/Professor/LHC staff)
- Supporting Actor: Database, Mail Client

Exceptions -

- Database connection failure while saving the request.
- Email notification fails due to SMTP service issues.
- Validation errors in the request form.

Includes -

- U3: Validate User Login.
- U8: Trigger Email Notification.

Notes/Issues -

- Need to ensure the database is available and functional at all times.
- SMTP service configuration must be robust and tested to handle high traffic.

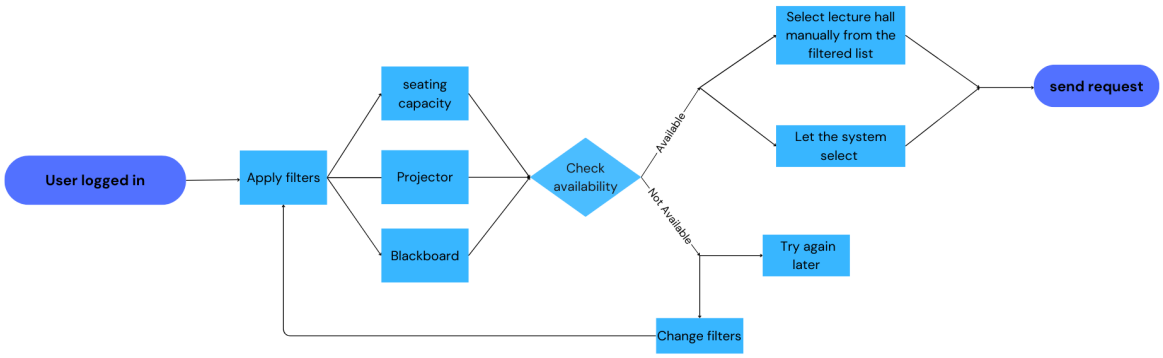

Figure 8: Use Case 1

3.3.2 Use Case #2: Authority Verifies Request (U2)

Author – Bhavya, Divyesh

Purpose - To allow authorities to review and approve or reject user-submitted requests.

Requirements Traceability –

- R4: Authorities must receive a notification for each new request.
- R5: The system must display all relevant details of the request for verification.
- R6: The approval or rejection decision must be recorded in the database.

Priority - High – Ensures smooth processing of user requests.

Preconditions -

- The request must exist in the system.
- The authority must be logged in securely.

Post conditions -

- The approval/rejection decision is recorded in the database.
- The user is notified of the decision.

Actors –

- Primary Actor: Authority

- Supporting Actor: Database, Mail Client

Exceptions -

- Database errors during decision recording.
- Failure in notifying the user.

Includes -

- U5: Notify User of Decision

Notes/Issues -

- Ensure decision-making is logged for audit purposes.
- Handle cases where authorities fail to act on notifications.

3.3.3 Use Case #3: User Login (U3)

Author – Atharv, Devansh

Purpose - To allow users to request login to the portal on the web or a mobile application to access the portal. Login be granted based on valid credentials.

Requirements Traceability –

- R7: Users must log in using valid credentials.
- R8: The system must differentiate roles (User, Office Staff, Authority) to grant appropriate access.

Priority - High – Ensures secure access and data protection.

Preconditions -

- The user must have valid credentials (username and password).
- The database must store user credentials securely.

Post conditions -

- The user is authenticated and granted access to role-specific features.
- Login activity is logged for security purposes.

Actors –

- Primary Actor: User (Club-coordinator/Professor/LHC staff)
- Supporting Actor: Database

Exceptions -

- Invalid credentials entered by the user.
- Database unavailability during login.
- Account locked due to multiple failed login attempts.

Includes -

- U6: Validate Credentials
- U7: Log Activity

Notes/Issues -

- Ensure encryption of passwords in the database.
- Implement a mechanism for password recovery.

3.3.4 Use Case #4: Generate Reports (U4)

Author – Avantika, Aaradhya

Purpose - To generate and export reports on user requests, statuses, and actions for analysis and auditing.

Requirements Traceability – R9: The system must allow exporting data in various formats (e.g., CSV, PDF). R10: Reports must be role-specific (e.g., user-specific, overall statistics). **Priority** - Low – Not critical for initial deployment but valuable for audits and reviews.

Preconditions -

- The data for reports must exist in the database.
- The user requesting the report must have appropriate access.

Post conditions -

- A downloadable report is generated and made available to the user.

Actors –

- Primary Actor: Office Staff, Authority
- Supporting Actor: Database

Exceptions -

- Data query errors due to database inconsistencies.
- Large report sizes leading to processing delays.

Includes -

- U9: Fetch Data for Report

Notes/Issues -

- Ensure that sensitive information is excluded from reports based on roles.
- Optimize database queries to handle large datasets efficiently.

3.3.5 Use Case #5: User Feedback (U5)

Author – Areen, Daksh

Purpose - To allow users to provide feedback on their experience with the application, report issues, or suggest improvements. The feedback will be stored in the database and accessible to the development team for review.

Requirements Traceability –

- R11: The system must provide a feedback form accessible to all users.
- R12: Feedback must be stored in the database for analysis.
- R13: Feedback submissions must include optional user contact details for follow-up.

Priority - Medium – Improves the application through user insights but is not critical for initial deployment.

Preconditions -

- The user must be logged into the application.
- The feedback form must be available and functional.

Post conditions -

- The feedback is stored in the database.
- Optional notification is sent to the development team for high-priority feedback (e.g., bugs or critical issues).

Actors –

- Primary Actor: User (Club-coordinator/Professor/LHC staff)
- Supporting Actor: Database

Exceptions -

- Database errors while saving feedback.
- Failure to notify the development team about critical feedback.

Includes -

- U11: Store Feedback in Database
- U12: Notify Development Team

Notes/Issues -

- Ensure the feedback form is simple and intuitive.
- Classify feedback based on priority (e.g., bug reports, suggestions, general feedback).

3.3.6 Use Case #6: Notify User of Decision (U6)

Author – Avantika, Aaradhya

Purpose - To inform the user about the approval or rejection of their submitted request via email or in-app notifications.

Requirements Traceability –

- R14: The system must send notifications to users for all decisions made by authorities.
- R15: Notifications must include details of the decision and any relevant remarks.

Priority - High – Critical for ensuring users are informed about the status of their requests.

Preconditions -

- A decision (approval or rejection) must be recorded in the database.
- The user must have provided valid contact information.

Post conditions -

- The user is notified of the decision.
- The notification is logged for audit purposes.

Actors –

- Primary Actor: Authority
- Supporting Actor: Database, Mail Client, Notification System

Exceptions -

- Email notification fails due to SMTP service issues.
- In-app notification fails due to server errors.

Includes -

- Log Notification in Database
- U14: Trigger Email/Notification

Notes/Issues -

- Ensure email templates are clear and user-friendly.

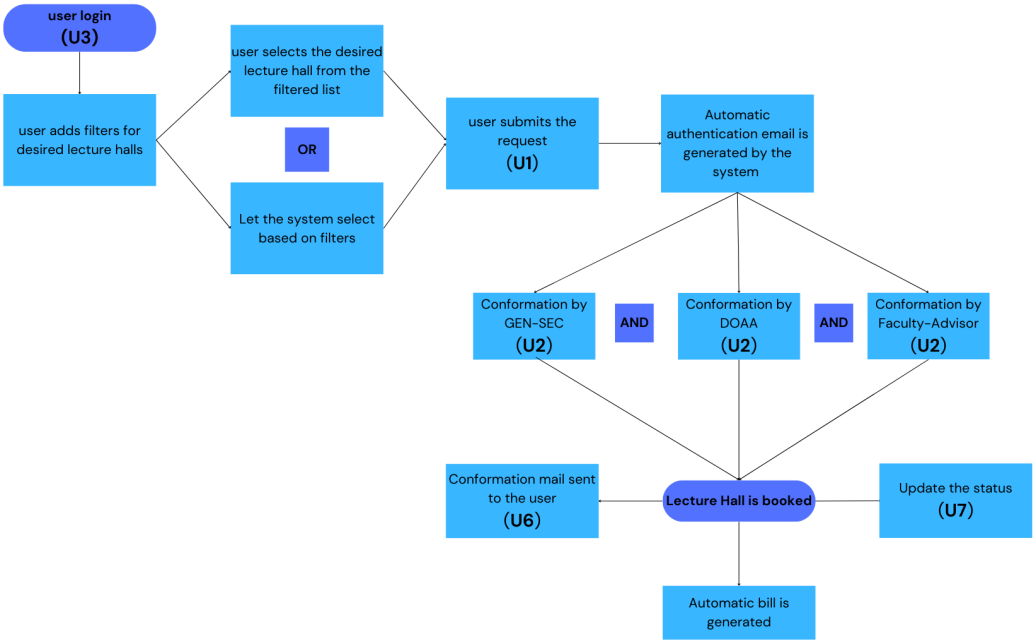

Figure 9: Entire Use Case Diagram

4 Other Non-functional Requirements

4.1 Performance Requirements

- **System Response Time:** All UI interactions, such as clicking buttons, navigating menus, and searching for halls, must have a response time of **under 5 seconds** to maintain a smooth user experience.
- **Concurrent users:** The system should support **at least 20 concurrent users** without degradation in performance.
- **Real Time Updates:** The system should have real-time updates or near-real time of the booking confirmations in progress. The booking confirmation can be confirmed or rejected by authority within **72 hours** of booking.
- **Scalability:** The system should scale seamlessly to support additional campuses or departments, increasing the number of users and lecture halls by **50%** without requiring significant architectural changes.
- **Custom Reports:** On-demand custom reports, such as daily usage summaries or availability statistics, should generate and display within **5 seconds** for datasets containing up to **2,000 records**.
- **Throughput:** The system must handle at least **5 successful booking transactions per minute** during peak usage, ensuring a seamless experience for all users.

4.2 Safety and Security Requirements

- **User Education:** All users of the software must authenticate themselves using unique credentials (username and password) before accessing the system.
- **Session Management:** User sessions should automatically time out after 15 minutes of inactivity to prevent unauthorized access from unattended devices ensuring secure session handling to prevent session hijacking.
- **Safety Features:** Provide a confirmation step before finalizing any booking, cancellation, or deletion to prevent accidental actions. Include mechanisms to prevent double booking of lecture halls.
- **Security Warnings:** Notify administrators of any suspicious activities, such as multiple failed login attempts or attempts to access restricted areas of the system.
- **Access Control:**Sensitive administrative functions, such as editing or deleting bookings, should be restricted to authorized personnel.
- **Compliance with Institutional Policies:** Align all safety and security measures with the institution's IT policies and guidelines.
- **Privacy Requirements:** Personal data such as email addresses and booking history should only be accessible to the user and authorized administrators.

4.3 Software Quality Attributes

4.3.1 Reliability

Requirement:

- The lecture hall booking system shall ensure availability during working hours (8:00 AM to 10:00 PM) to ensure uninterrupted access for users. Any downtime for maintenance must be scheduled outside these hours and communicated to users at least 24 hours in advance.

Plan to achieve:

  - Implement automated monitoring and alert systems to detect and resolve issues proactively.
  - Use redundant database backups to recover quickly in case of failures.

4.3.2 Adaptability

Requirement:

- The lecture hall booking system shall support the addition of new lecture halls, campuses, or user roles (e.g., event coordinators) without requiring significant code changes. The system should allow administrators to configure these settings via a web-based dashboard.

Plan to achieve:

  - Use a modular design and database schema that supports dynamic addition of entities (e.g., lecture halls, campuses, user roles).
  - Implement a role-based access control (RBAC) mechanism to handle new roles easily.

Appendix A: Data Dictionary

Item	Type	Description	Operations
Constants			
MAX_HALLS	Integer	Maximum number of lecture halls available for booking	System will not allow booking beyond this limit
MAX_BK_DUR	Integer	Maximum duration allowed for booking a lecture hall (in hours)	Typically 2 hours; system rejects bookings beyond this duration
SYSTEM_TIMEZONE	String	The timezone in which the booking system operates	”Asia/Kolkata” or as per local setting
price	Integer	Variables assigning prices to each LH and equipment	System will use this to generate invoice
AUTO_REJ_TIME	integer	The time in hours after the booking will be auto rejected.	typically 50-48 hours
State Variables			
booking_status	String	Current status of the booking (e.g., ”Available”, ”Booked”, ”Pending”)	Operations: Update status when booking is made or canceled
user_role	string	Role or designation of each user, normal_user , admin, club, sop, professor, etc.	Users are give special permission and validated based on their role
hall_id	Integer	Unique identifier for each lecture hall	Operations: Generated during system setup, used to track bookings
booked_by	String	Name or ID of the user who has booked the lecture hall	Operations: Updated when a booking is made
booking_time	DateTime	Date and time of booking	Operations: Set when a user initiates a booking

Item	Type	Description	Operations
booking_end_time	DateTime	End time of the booking	Operations: Set based on the start time and maximum booking duration
activity_type	string	Type of activity the user will be using the LH for (e.g., ”exam”, ”placement” , ”session”, ”workshop” ,”lecture”)	Operations: User selects while filling the request form
Inputs			
user_id	Integer	ID of the user who is making the booking request	Operations: Verified against system user database
req_hall_id	Integer	LH ID that the user wants to book	Operations: Checked for availability in Database.
req_start_time	DateTime	Requested start time for booking	Operations: Checked for availability against other bookings
req_end_time	DateTime	Requested end time for booking	Operations: Checked for availability against other bookings
Outputs			
bk_confirmation	String	Confirmation message or booking reference ID	Operations: Sent to the user upon successful booking
error_message	String	Error or failure message (e.g., ”Hall unavailable”, ”Invalid time slot”)	Operations: Displayed to user if booking fails
booking_history	List	A list of past bookings for a specific user or hall	Operations: Displayed when a user or admin checks booking history
invoice_number	integer	Total cost of booking include equipment charge	Operations: Displayed to user when downloading invoice

Item	Type	Description	Operations
Functional Requirements			
User Authentication	Function	Verify if the user is authorized to make a booking	Operations: System must validate the user ID and password
Hall Availability Check	Function	Ensure requested hall is available at the requested time	Operations: Check if no conflicting bookings exist for the requested time
Booking Request Approval	Function	Admin must approve the booking request before it becomes final	Operations: Admin reviews and confirms booking requests
Booking Confirmation	Function	System sends confirmation upon successful booking	Operations: Email or SMS notification to the user
Booking Cancellation	Function	Users or admin can cancel bookings	Operations: Update the booking status, free the hall
Invoice generation	Function	System automatically generates invoice	Operations: Includes the overall cost of the booking.

Appendix B: Group Log

Since the beginning of the project, our entire team has been very enthusiastic. We have formed a Whatsapp group for effective communication among the team members.

Meeting minutes	**Agenda**
8 Jan 7:00 to 8:10 pm	We discussed a lot of different ideas and thought of different options.
	We discussed different ideas with Prof Indranil Saha and finalized an automated Lecture hall booking portal.
17 Jan 9:00 to 9:50 pm	planned out the draft of the SRS, discussed other logistics and deadlines, and started the work with full enthusiasm.
17 - 19 Jan	Worked out the draft of the SRS. We aligned our ideas and made a proper flow for the work.
22 Jan	We met with the TA. We discussed the issues faced by the team.
23 Jan 9:00 to 10:30 pm	We were done with most of the raw work and went through the work done other members to avoid errors.and we went through the complete SRS once and discussed about the issues spotted
24 Jan	Gave the final touch to the SRS for the submission.